EXTENSIONS: Issue #1
For the PRO/FILE 2068 user

Welcome to the first issue of EXTENSIONS, an unofficial and irregularly printed newsletter for PRO/FILE 2068 owners. As of now I have another issue planned which will feature an update routine. The routine will permit you to change the data on any line to different data. For example: On line 6 you list a product code for your line of Whiz-bang pliers. All your needle-nose pliers have a code starting with N but your company wishes to change this to NNP. With the update routine this would be done automatically in every appropriate file! Also planned is a more versatile EDIT TO NEXT routine, repetitive file additions, starting a search from any point in your files, and more. Anyway, that's in the next issue which is now available for $6.00. Now on with this issue.

The changes which follow sometimes require that you have previously added options from the manual or added the machine code sort routine from Tom Woods' BREAKTHROUGH newsletter (I strongly suggest you get it). I recommend that you add all the improvements from the manual (you don't need to add the ALPHABETIZE MORE THAN FIVE CHARACTERS or the DISABLE AUTO REPEAT routines if you don't want to or the additions for a full size printer). If you do everything listed in the manual and the BREAKTHROUGH newsletter as well as everything in this newsletter and my next one, you will still have all the capacity of the original version for your files!

Also, keep copies of old versions with data for a while since you may make an error when making these changes and the resulting bug could result in the permanent loss of data.

MAKING BACKUPS

Before we get too involved with making changes let's provide a simple way to make backups of your master program (that is an empty basic program and machine code).

Break into the program (see instructions in the Pro/file manual-page 97) and then add these two lines:

```
9990 CLEAR: SAVE "PRO/FILE" LINE
 9991: BEEP .5,30: SAVE "p/f"COD
E 63488,2046: BEEP .5,30: VERIFY
 "": VERIFY ""CODE : BEEP .5,40:
 GO TO 9996

9991 BORDER Ø: PAPER Ø: INK 7: C
LEAR 63487: LOAD ""CODE : CLS :
POKE 23658,8
```

WARNING: Later we will use many variables to save memory and you will change most lines to take advantage of them. However, do not use any variables in either of the above lines because the first thing that happens on execution is that all variables and data are erased!

Now, make sure you have a copy of your program and data before we use this to ever make a backup since all data will be erased. When that is done just type RUN 9990 or GO TO 9990 after you have started your tape. When you hear a beep, quickly press ENTER to save the machine code. When the next beep sounds you are done and the program will wait for you to rewind the tape and play it back for verification.

You can now create a new file or load in another or just break in again and repeat the process to make another master copy. Since these two lines remain you can make new master copies at any time.

When you load in one of these copies it will auto-run at 9991 which loads in the machine code and then it restarts as the original does.

HOW TO TRANSFER DATA TO NEW VERSIONS

If you have more than one tape of PRO/FILE, you are aware of the problem of making improvements since they must be typed into each copy individually. To simplify this, follow these instructions (to use my SAVE routine described later you need to do this anyway so do it now).

1. Break into the program as described in the manual.
2. As a direct command type in SAVE F$CODE PEEK 23627+256*PEEK 23628+6,P
3. Start your recorder and press ENTER to save the data.
4. Rewind the tape, type VERIFY ""CODE and press ENTER and start your tape to verify.
5. If it does not verify then try again, otherwise type PRINT P and ENTER. Write the answer down!
6. Repeat the process for each of your file tapes.

The only thing we have saved is raw data from D$. Since we skipped over the bytes which describe the length of D$ and only saved the bytes actually used for files (the value of P), you can move data from programs having different lengths for D$ to your new program. The only requirement is that the length of D$ in the new version must be at least 10 bytes longer than the value of P if you use my SAVE routine or equal to P without it. If you reduced the size of D$ to make improvements from the manual you can use my memory saving devices described later to restore it back to its full size or even more.

Put those data tapes away for now. Later, after we have added our improvements, I'll show how to transfer them in.

FIX FOR TALLY ROUTINE

The following is only for those who have added the TALLY routine from the manual AND the MACHINE CODE SORT routine from Tom Woods' Breakthrough newsletter. When the mc sort is added to a program containing the tally function two bugs appear. The first and most important is that you can no longer do an auto search with the printer output set to "Y". Instead of automatically going through each matching file, the program stops at each one. The other bug only occurs if you do a search with the Tally in effect and immediately follow it with the mc sort. It won't work! No damage seems to be done however and if you try to sort again it will work. To solve both these problems add or change the following lines:

```
  3 LET J$="": LET A=Ø: LET X$=
"": LET T=Ø: LET TØ=Ø

7229 IF J$="T" THEN CLS : GO TO
1049+(Y$="C")

9210 IF Y AND J$="T" THEN GO TO
150
```

SAVING MEMORY

The use of variables, the VAL function, and the CODE function will allow you to save almost 2000 bytes as done below. Actually, more could be saved with more variables, but I limited it to the most useful values.

First change line 9996:

```
9996 DIM d$(VAL "28020"): LET p=
VAL "20": LET O=NOT PI: LET U=SG
N PI: LET G=VAL "7": LET J=VAL "
15": LET F=VAL "4": LET d$( TO P
)="*SEARCH IS COMPLETE*": LET a$
="ALL": LET c$="": FOR x=U TO J:
 LET c$=c$+CHR$ x: NEXT x: LET s
=O
```

The letter "O" stands for the number zero. Instead of using LET O=Ø which takes 9 bytes I use LET O=NOT PI which takes just 5 bytes. For the same reason I use LET U=SGN PI instead of LET U=1. Another possibility would be to use INT PI instead of the number 3. You can use these directly in your own programs for considerable savings. Using VAL instead of just a number in the other variables saves 2 bytes each. If you are adding this to a file you are presently using, you will need to enter O=Ø, U=1, F=4, G=7, and J=15 as direct commands as well.

Now go through each line of the program and change every reference to the number Ø to the letter O, each number 1 to the letter U, etc. Also replace the number 2 with U+U, 3 with F-U or INT PI, 5 with F+U, 6 with G-U, 8 with G+U, 11 with F+G, 14 with G+G, 16 with J+U, and 30 with J+J. It is a tedious task but it will provide BIG savings which allows you to maintain full capacity even with all these additions. Also, you can replace other numbers with CODES. For example, the number 32 is the code for a space so the number 32 can be replaced with CODE " ". For example: Instead of FOR X=1 TO 32 you would use FOR X=U TO CODE " " which saves 8 bytes. Using CODE does make the program more difficult to follow and the savings are smaller so you may not want to do it. However, you should use VAL which can be used with any number. Instead of RANDOMIZE USR 64731 just use RANDOMIZE USER VAL "64731". Each time you use VAL you save 2 bytes. With so many numbers in the program it really mounts up. Using a variable saves up to 6 bytes (but usually 5). Be careful about confusing a lower case "L", that is "l", with the number 1.

The disadvantage of using these techniques is that the program slows down. In a basic program this could become obvious, but since most of the sections of PRO/FILE are in machine code the difference in speed is negligible. Without a disk drive for fast loading and saving, I feel it is much more important to have as much capacity as possible. An exception is in lines 5010 through 5103. These lines handle the process of editing new or old files and the use of variables, etc will slightly reduce the speed you can type at so it may be wise to leave those lines alone. Also remember to leave 9990 and 9991 as is.

Please note that most of the program lines printed in this newsletter are written without these techniques in case you choose not to use them. Of course, if you don't use them you will have to reduce the size of D$ considerably. If you are using them then adjust each line accordingly.

COMPACT MENUS AND MORE SAVINGS

The display menu and edit menu each consist of several sections. I prefer to see as many options at one time as possible. Also, if you have a long search command (more than one line long) you will run into a problem with the screen SCROLL. These changes solve both situations and save about 350 bytes as well.

```
 102 IF X$="DEFP" THEN PRINT AT
14,0;: PLOT 120,70: GO SUB 6500:
 INK 0: GO SUB 9830: INK 7: PRIN
T AT 9,7;A$; INK 0;E$(1);E$(1);E
$(1): GO TO 50

1040 PRINT AT 20,0;X$

1047 PRINT AT 20,0;X$

1050 GO SUB 9100: PAPER 5: INK 0
: PRINT AT 16,1;"Press ENTER to
continue","C)opy  D)elete  E)dit
  A)dd fileG)oto menu        R)es
tart searchM)ore info   TYPE SEA
RCH COMMAND": INPUT "OPTION? ";Y
$

1051 IF Y$="M" THEN PAPER 0: INK
 5: PRINT AT 16,0;: GO SUB 9850:
 INPUT "OPTION? ";Y$: IF Y$="M"
THEN GO TO 1050

1070 IF Y$="G" THEN GO TO 1

5004 INK 1: PLOT 0,48: PAPER 6:
DRAW 255,0: INK 0: PRINT AT 16,0
; PAPER 1; INK 7;"ENTER:Next lin
e or use SHIFT and1:Switch Inser
t/Over 2:Caps Lock3:Line Delete
      4:Line Erase5-8:Move Curso
r      9:More Info";AT L,C;: LET
 M=0

5115 IF M=0 THEN PRINT BRIGHT 1;
 PAPER 1;AT 17,0;"SYMBOL SHIFT:L
ine Insert","STOP:Close File",,:
 LET M=-1: GO TO 5040

5116 IF M=-1 THEN PRINT AT 16,0;
: GO SUB 9850: LET M=1: GO TO 50
40

9850 PRINT "FILE: "; INK 7;F$; I
NK 3;"     ORDER: "; INK 7;S, IN
K 3;"OPEN: "; INK 7;LEN D$-P;: P
RINT " bytes";TAB 0;: PRINT "FOR
MAT:"; INK 7;A$,,,TAB 0;: RETURN
```

Remember that you now use "G" to return from the display menu to the main menu instead of "N"!

THREE WAY SAVE WITH VERIFY

On the original PRO/FILE you could only save the data with the basic part of the program. It required you to load in a master program at first to get the machine code section but didn't have the speed advantage of only saving the data. This section allows you to save ALL (including machine code so you don't need a separate master), or the basic with data as is done on the original, or just the DATA itself. In fact, it doesn't even save all he data, but essentially only the part of D$ actually filled. If you only used a few thousand bytes of D$ then that's all you save. This cuts way down on loading and saving time. This is done by saving the data as CODE instead of as DATA.

To do this I changed the order of the initial set-up so that P=20 is first and DIM D$(28020) is second. The reason D$ was originally first was simply so that it would be in a known position (the start of the variable space). Putting P first still leaves D$ in a known position because any numeric variable takes 5 bytes plus one for each letter in the name. Therefore D$ will now be at the start of the variables + 6. When put in before any string variables, your numeric variables, DIM strings and arrays do not move around because their length never changes. However, string variables (the undimensioned type) can change length and so move around.

These additions should be done after you have saved your files as described at the beginning of this newsletter. Then break in and CLEAR the program.

```
   60 Same as for verify in PRO/FILE manual.

  107 IF X$="SAVE" THEN GO TO 8200

 1080 IF PEEK 64026+256*PEEK 6402
 7<>PEEK 23627+256*PEEK 23628+12
 AND (Y$="D" OR Y$="E") THEN GO T
 O 4000

 5505 PRINT AT 16,1;"   LOAD DATA
  ONLY? (Y OR N)   ": PAUSE Ø: LE
 T X$=INKEY$: IF X$="Y" THEN LOAD
  ""CODE PEEK 23627+256*PEEK 2362
 8: LET F$=D$(P+1 TO P+10): BEEP
 .5,40: GO TO 1

 5506 IF X$<>"N" THEN GO TO 5505

 8000 Same as for verify in PRO/FILE manual.

 8100 POKE 23730,255: POKE 23731,
 247: IF PEEK 64268<>205 THEN LOA
 D ""CODE 63488,2046

 8110 BEEP .5,30: BORDER Ø: PAPER
  Ø: INK 7: GO TO 1

 8200 CLS : PRINT AT 7,Ø;"SAVE:
 A)LL"''TAB 7;"B)ASIC WITH DATA"'
 'TAB 7;"D)ATA ONLY": PAUSE Ø: LE
 T X$=INKEY$: IF X$<>"A" AND X$<>
 "B" AND X$<>"D" THEN GO TO 8200
```

```
8205 CLS: ON ERR GO TO 8000: IF
X$="D" THEN LET D$(P+1 TO P+10)=
F$: SAVE F$CODE PEEK 23627+256*P
EEK 23628,P+22: GO TO 8220

8210 SAVE F$ LINE 8100: IF X$="A
" THEN BEEP .5,30: SAVE F$CODE 6
3488,2046

8220 BEEP .5,30: PRINT AT 20,Ø;"
REWIND TO VERIFY/BREAK TO STOP "
: IF X$<>"D" THEN VERIFY F$

8225 IF X$<>"B" THEN VERIFY ""CO
DE

8230 BEEP .5,30: GO TO 1

9850 PRINT "FILE: "; INK 7;F$; I
NK 3;"     ORDER: "; INK 7;S, IN
K 3;"OPEN: "; INK 7;LEN D$-P-10;
: PRINT " bytes";TAB Ø;: PRINT "
FORMAT:"; INK 7;A$,,, TAB Ø;: RE
TURN

9996 LET P=20: DIM D$(28020)   etc

9997 PRINT AT 19,2; INK 7;"Press
 ""C"" to CREATE a new file or "
"L"" to LOAD an existing one": I
NPUT Y$: PRINT AT 19,Ø;D$(100 TO
 164): IF Y$="L" THEN GO TO 5505
```

Now as a direct command enter POKE 64095,12. If you have added the machine code sort you must also POKE 64754,12. If you add the mc sort later be sure to make the poke after you add it but before you use it!

When you SAVE ALL you will actually be making two saves. The second will save the machine code. After the first part is saved a beep will sound. You must then press a key to save the second part. Another beep sounds when done.

When using the DATA save remember that you can only load data into a program if the length of D$ is the same for both. The only way to move data into a different size of D$ is described at the beginning of this newsletter for the saving instructions and at the end for loading the data back in.

There is one unusual aspect of the SAVE routine. If you load in an empty master tape (described later) and immediately opt to load a data tape you will not be able to find a file on the initial search (you will just get SEARCH IS COMPLETE"). If you then press "R" to restart the search or enter another search command everything works fine. If you loaded a master tape and created a new file first and then load a data tape it also works fine. I can find no reason for this and have suffered no ill effects from it, but it is very puzzling. Anyway, it only occurs the first time you load it in. If it really bothers you you can add a line 1 LET T1=0. This solves the problem for me but for no logical reason I can see.

AVOID PRINTING "SEARCH IS COMPLETE" OR DIRECTORY FILES ON AUTO SEARCHES

Being a band director, I often made printouts of my students by instrument. First I would print all the flute players, then clarinets, etc. However, I always had SEARCH IS COMPLETE printed in between each section. Also, I use a number of codes in my files, so to help me keep track of their meaning I make directory files which list and explain each one. Unfortunately, if I do a printout of all school owned instruments (I would search for ";/") I would also printout the directory which contains that code! The following line solves both problems.

```
7203 IF B=P OR E$(1, TO 2)="##"
THEN GO TO 7229
```

The only time B=P is when SEARCH IS COMPLETE is on the screen and so the print routine is skipped over. If you start any directory files you don't want auto printed with ## as the first 2 characters on line 1 they will be by-passed as well. Note, this only affects auto printing. You can still print them out as individual files. Also, any time you want to see your directories just search for ##

SET CAPS LOCK AUTOMATICALLY

It is not a big deal, but I don't like having to set caps each time I load the program (I rarely use lower case). Adjust line 8100.

```
8100 POKE 23730,255: POKE 23731,
247: POKE 23658,8: IF PEEK 64268
()205 THEN LOAD ""CODE 63488,204
6
```

If you will always be loading a master copy first as I described earlier you don't need to make the above change, but if you sometimes use my SAVE A)ll routine then you will.

POKE 23658,8 can be used in any program to set caps. Use zero to set lower case. The same instruction can be inserted elsewhere if it suited you better (at start of line 100 it would reset caps each time you returned to the main menu.

ADDING A SWITCHABLE BEEP

This key click can be turned on or off by entering "T" as a search command from the main menu (I use "T" for tone instead of "C" for click to avoid any confusion with C)opy). The keys will click on any key press, not just on the ADD or EDIT routines.

5050 If you used the key click method described in the manual you must first delete line 5050!

```
  3 POKE 23609,E: LET J$="": LE
T A=Ø: LET X$="": LET T=Ø: LET T
Ø=Ø

 108 IF X$="T" THEN LET E=ABS (E
-10): GO TO 1

5020 LET Y$=INKEY$: BEEP E/1000,
40: FLASH Ø: PRINT AT L,C;SCREEN
$ (L,C): IF CODE Y$(16 THEN GO T
O 5100+CODE Y$
```

```
9997 LET E=10: PRINT AT 19,2; IN
K 7;"Press ""C"" to CREATE a new
file  or ""L"" to LOAD an existi
ng one": INPUT Y$: PRINT AT 19,Ø
;D$(100 TO 164): IF Y$="L" THEN
GO TO 5505
```

MISC

At the end of line 5040 there is a delay loop in some versions which I have found to be unnecessary in my experience. You can simply delete the section which reads

```
: FOR X=1 TO 5: NEXT X
```

Also, the routine for editing new or old files seems to work better with the following change:

```
5016 LET Y$=INKEY$: IF Y$)=" STE
P " THEN ON ERR RESET: FLASH Ø:
GO TO 6000
```

```
5020 BEEP E/1000,40: FLASH Ø: PR
INT AT L,C;SCREEN$ (L,C): IF COD
E Y$(16 THEN GO TO 5100+CODE Y$
```

That should cut down on the problem of hitting a key and having nothing appear on the ADD/EDIT screen.

FINALLY!!!

Now you can transfer any of those data tapes in as previously described. Just follow these instructions:

1. Have an improved copy of Pro/file in your computer. If you have just loaded in a master copy choose the CREATE option and then enter the name of the file data you will be transporting over. If this is not a freshly loaded master, then first break in and enter RUN 9996.
2. If you are using my SAVE options then break into the program and enter LOAD "" CODE PEEK 23627+256*PEEK 23628+12 and play in the data tape. If you are not using my save routines then do the same thing but change the +12 to +6.
3. When loading is complete, enter LET P=whatever you wrote down for this data. At this point we have given the proper name to the file, transferred the actual data, and reset the value of P. That's all we need so:
4. Enter GO TO 1.

You should already have made backup master copies of the program so now you can just save the data if you put in my save routines or save one of the other ways if you prefer. When done go back and transfer another data tape if any. Keep in mind that this process allows you to put your files in a fully expanded D$ array even if you had previously reduced capacity when adding options from the manual.

Remember to keep previous versions of your files somewhere in case you made an error. I have found from experience that such bugs often don't appear for quite some time.

That's it for this issue. The next will be even more useful!

Robert C Fischer, 221 Scoggins St, Summerville, Ga 30747